

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
15 September 2005 (15.09.2005)

PCT

(10) International Publication Number
WO 2005/084233 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2005/006171

(22) International Filing Date: 25 February 2005 (25.02.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/789,665 27 February 2004 (27.02.2004) US

(71) Applicant (for all designated States except US): **FORT POINT PARTNERS, INC.** [US/US]; Pier One, Bay Three, San Francisco, CA 94111 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ROCHE, Matthew** [US/US]; 131 Lake Street, San Francisco, CA 94118 (US). **PEARSON, Benjamin** [US/US]; 956 Grizzly Peak, Berkeley, CA 94708 (US).

(74) Agent: **LAM, Wilfred**; Innovation Management Sciences, P.O. Box 1169, Los Altos, CA 94023-1169 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

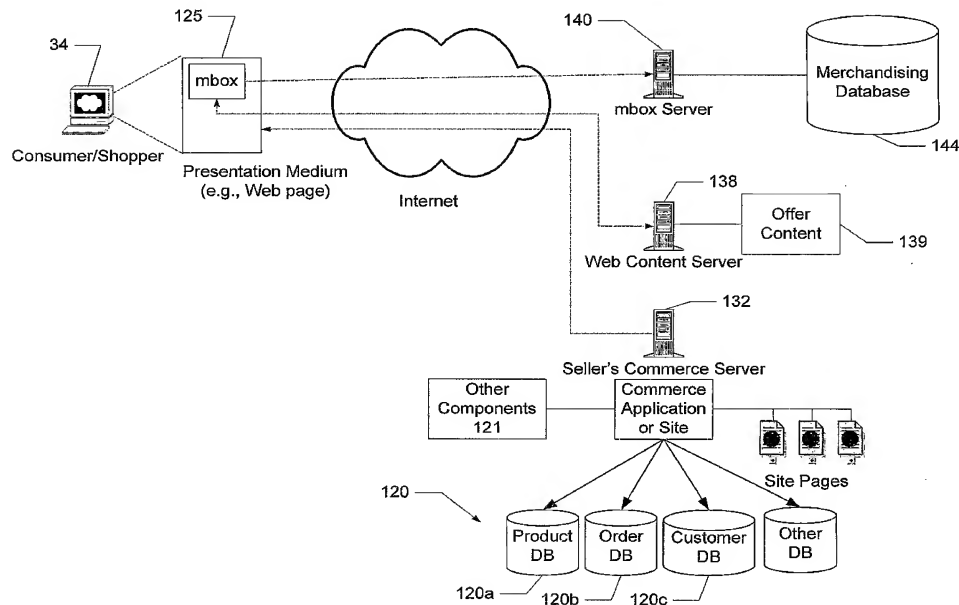
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR COLLECTING ONLINE MERCHANDISING DATA



(57) Abstract: A method and system for automating electronic merchandising and product recommendation by creating a third party merchandising product database populated with data obtained directly at the point of sale. According to one embodiment, collection or direct importation of such data from an existing physical source or a product or merchandising database, which can be complex and costly, is not required. As a result, the cost of creating a usable merchandising system and database is reduced.

WO 2005/084233 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND SYSTEM FOR COLLECTING ONLINE MERCHANDISING DATA

COPYRIGHT NOTICE

5

[0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise
10 reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

15 [0002] The present invention relates to electronic commerce. More specifically, the invention relates to a method and system for populating a merchandising product database with product data obtained at the point of presentation or sale.

20 BACKGROUND OF THE INVENTION

[0003] Internet and related technologies have radically altered how transactions are conducted between businesses and between consumers and suppliers. Online sellers who use these technologies are capable of accepting orders for products and
25 services without manual intervention by the seller. In addition, the online selling environment allows businesses to collect data that can be used to better position and promote their products and improve profitability and volume of sales.

[0004] Using this data, an online seller can increase the likelihood of selling to a customer by offering product suggestions based on how that shopper or other related
30 shoppers have browsed and transacted with the business. In addition, sellers have the opportunity to change the display, description, and location of product suggestions

dynamically based on the information available during the online selling process. The optimization of product presentation through recommendations, pricing, product description, assortment, signage, and other means is referred to as online merchandising.

5 [0005] The data used by online merchandising is generally referred to as merchandising product data. This merchandising product data generally includes product information (such as product name, SKU number, price, description, and photograph, etc.), data about a product's presentation, assortment and packaging relative to other products or services, and data that is derived from customer buying
10 and browsing activities, such as number of times a product is viewed, number of times the product is purchased, number of click-throughs, buyer segment association, other products associated with the product by virtue of being purchased together, etc.

[0006] Currently, collecting different types of data into a database that can be used for online merchandising purposes requires either rebuilding the seller's online
15 systems or performing very costly direct integration of seller's product databases with third party software and services. The cost of rebuilding an online system or performing direct database integration often exceeds the potential sales gain for the seller.

[0007] Moreover, the technical complexity of these systems demands
20 professional engineering support for relatively simple online merchandising efforts, greatly increasing the cost and decreasing the speed of implementation.

[0008] Accordingly, there exists a need for a method and system that significantly reduces the complexity and cost of creating a usable merchandising product database. Additionally, there exists a need for a method and system that
25 minimizes integration with core customer database systems to reduce or remove costs and delays related to the requirement for engineering resources for database integrations or technical support.

30 **BRIEF DESCRIPTION OF THE DRAWINGS**

[0009] The invention will now be described with reference to the accompanying drawings which illustrate an example embodiment of the invention. Throughout the description, similar reference names may be used to identify similar elements.

[0010] FIG. 1 depicts an embodiment of the invention.

[0011] FIG. 2 depicts the runtime and administration components of an Mbox Server according to an embodiment of the invention.

5 [0012] FIG. 3 depicts an example product detail page having an mbox embedded therein, in accordance with an embodiment of the invention.

[0013] FIG. 4 depicts an example product detail page and an mbox displaying a product recommendation offer, in accordance with an embodiment of the invention.

[0014] FIGS. 5A-5B depict a flow diagram for a process of setting up an Mbox Server in accordance with an embodiment of the invention.

10 [0015] FIG. 6 depicts a flow diagram for a process of selecting appropriate offer content to be displayed at an mbox, in accordance with an embodiment of the invention.

[0016] FIG. 7 depicts an example user interface of the Mbox Server through which a user can create and define parameters of a Merchandising Campaign.

15 [0017] FIG. 8 depicts an example user interface of the Mbox Server through which a user can select an automated merchandising element for an mbox.

[0018] FIG. 9 depicts an example computer system in which an embodiment invention can be implemented.

20 [0019] FIG. 10 depicts an example report summarizing the findings of an A/B Comparison Campaign according to an embodiment of the invention.

[0020] FIG. 11 depicts another example report summarizing the findings of an A/B Comparison Campaign according to another embodiment of the invention.

25 **DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS**

[0021] Various features of the invention, including specific implementations thereof, will now be described. Throughout the description, the term “product” will be used to refer generally to a saleable unit, whether a physical or digital object, a
30 service, or data that may have a range of attributes (e.g., style, size, color). The term “product” will also be used to refer generally to both something that may be purchased, and its record or description within a database (e.g., a mobile phone's description within a products database.) A more specific meaning may be implied by context. The term “item” is used in the same manner.

[0022] Throughout the description, the terms "product information" and "product data" will be used to refer generally to data that is inherently related to a product and its attributes, such as product name, SKU number, price, description, photograph, product family or style, etc. The term "customer response information" refers to data that is obtained or derived from customer buying and browsing activities, such as number of times a product is viewed, number of times the product is purchased, number of times conversions occurred (e.g., sales or click-throughs), buyer segment association, and other products associated with the displayed product by virtue of being purchased together, etc. The term "customer response information" may also refer to data that relates to promotional offers, and may also refer to shopper-specific data such as those indicating frequency of visit, impressions of other products by a shopper, etc. The term "customer response information" sometimes may refer to data that is not product information. The terms "merchandising product data" or "merchandising data" refer to data that may be relevant to or useful to merchandising activities and may include, but not limited to, product information, customer response information, and/or other data.

[0023] Throughout the description, the terms "point of presentation and/or sale", "point of presentation or sale" and "point of presentation" are used synonymously.

[0024] The various features of the invention set forth herein may be embodied within a wide range of different types of computer systems, including cable television systems, satellite television systems, wireless devices (including laptop computers, wireless telephones and personal digital assistants) and systems in which information is conveyed to users by synthesized voice. In addition, the output medium is demonstrated as a Web client, but could also include retail point of sale, videogame, television, or other digital media device. Thus, it should be understood that the HTML Web-site based implementations described herein illustrate just one type of system in which features of the invention may be used.

[0025] An embodiment of the present invention provides a method and system for automating electronic merchandising and product recommendation by creating a merchandising product database (or merchandising database) populated with data obtained at the point of presentation or sale. According to one embodiment, collection or direct importation of such data from an existing physical source, or from a product or merchandising database, which can be complex and costly, is not

required. As a result, the cost of creating a usable merchandising system and database is reduced.

[0026] According to one embodiment of the invention, the point of presentation or sale may refer to the output medium of an interactive catalog (e.g., a Web-page).

5 According to this embodiment, merchandising product data (including product information) is collected directly from the output medium of the interactive catalog. The merchandising product data is then stored within a database, which may not have any prior data about the product(s) displayed.

[0027] According to another embodiment of the invention, the point of
10 presentation or sale may refer to the time at which products are presented. According to one embodiment, the invention collects merchandising data when the customer browses an interactive catalog. Thus, merchandising data is collected contemporaneously with product presentation or sale regardless of where the merchandising data is collected. For instance, the merchandising data may be
15 collected from the seller's server that produces Web-pages. The merchandising product data is then stored within a database, which may not have any prior data about the product(s) displayed.

[0028] Some advantages of collecting merchandising data at the point of presentation/sale are as follows:

- 20 • Because of the nature of how sellers manage their online selling systems, the seller's online site often has more current and effective merchandising data than the seller's internal databases.
- According to an embodiment, the actual coding required to collect merchandising data at the point of presentation may be substantially
25 simpler than traditional data integration of product databases.
- According to an embodiment, the coding may be implemented by lower-skilled employees of the seller.
- According to an embodiment, the invention may collect merchandising data at a point where product data and customer buying and browsing data are both available, simplifying their combination.
30
- According to an embodiment, the invention may have an implicit scoring or "culling" ability as only those products that are viewed by the seller's customer will enter the system. Products that are not viewed or purchased will not be present in the merchandising database.

- According to an embodiment, the structure of the seller's product databases may be independent of the structure of the merchandising database. Thus, the seller may be able to freely modify/upgrade the product databases without having to modify the structure of the merchandising database.
- According to an embodiment, collecting merchandising data at a point of presentation allows a merchandising database to be managed by a third party, which in turn may further reduce the cost of implementing automated merchandising mechanisms.

10 **[0029]** The invention further provides a method and system that dynamically modifies interactive catalogs according to the collected merchandising product data. The invention enables online sellers to advertise, promote, and organize the sales of particular products using merchandising data collected at the point of presentation or sale. Because both product information and customer response information is
15 collected from the point of presentation/sale, there is no need to rebuild the seller's existing e-business systems or to perform direct integrations of the seller's existing source product database with third party software and services to obtain such merchandising data.

20 **[0030]** Consider an online seller that has implemented an embodiment of the invention. When the Web-pages of the online seller are rendered, product information and customer response information associated with the displayed products is collected and stored within a merchandising database. The collected merchandising data can be analyzed to produce business insights usable by a merchandiser to organize and promote the products more effectively.

25 **[0031]** For example, based on the merchandising data, the merchandiser may determine that product X has a higher conversion rate when displayed with product Y than when it is displayed with product Z. Then, the merchandiser may make changes to the Web-pages such that product X is displayed with product Y more often than with product Z to enhance sales.

30 **[0032]** According to one embodiment, the invention provides an automated merchandising mechanism that uses the collected merchandising data to automatically alter how products are merchandised. In this embodiment, the Web-pages of the online seller are configured such that Web-page organization, product presentation, and/or product description is finalized at the point of presentation or sale. In other

words, the automated merchandising mechanism automatically alters the way a product is merchandised, if necessary, when the Web-pages are displayed or requested. For example, the automated merchandising mechanism automatically selects a certain photograph to depict product X when the merchandising data indicates that conversion rate is the highest when that certain photograph is shown.

[0033] An embodiment of the invention may be implemented in the form of a software program, which may be provided by a third-party service provider on a subscription basis. In this embodiment, the subscriber business only needs to make minor modifications to their Web-pages to use the software program. Because the software program is not maintained by the subscriber business, and because only minor modifications to the existing Web-pages are needed, the cost of implementing the software can be minimized, and the software can be deployed in a very short time.

[0034] Referring now to **FIG. 1**, there is shown general components of a system according to an embodiment of the invention. As illustrated, the system may include: a computer or digital presentation device (including various wired and wireless devices suitable as an output of media data) **34**, a Seller E-Commerce Server **132** supporting product sales and controlled by a seller, an Offer Content Server **138** holding Offer Content **139** (e.g., product images, text, HTML, video, audio, or other media) used to present and promote products and promotions, and an Mbox Server **140**, which stores merchandising data in a Merchandising Product Database or Merchandising Database **144** and which may include engines for analyzing the collected data and producing recommendations and results.

[0035] The Seller's E-Commerce Server **132** further may include various databases **120** containing data that support product sales. These databases **120** typically include a product database **120a** for storing product information, an order database **120b** for storing purchase orders information, and a customer database **120c** for storing customer transaction histories.

[0036] Various other components the Seller's E-Commerce Server **132** are depicted as block **121** in **FIG. 1**. The components of block **121** include, for example, a search engine and associated database (not shown) for enabling users to interactively search the databases **120** for particular products. Also included within the block **121** are various order processing modules (not shown) for accepting and processing orders, a shopping cart module to add and remove items from the user's shopping carts, etc. The various processes of the Seller's E-Commerce Server **132**

may run, for example, on one or more Linux or Unix based workstations or physical servers.

[0037] A product data collection process according to an embodiment of the invention begins with a consumer/shopper operating a client (e.g., Microsoft Internet Explorer) on a presentation device 34 to generate a request for product information from the Seller's E-Commerce Server 132. Upon receiving the request, the Seller's E-Commerce Server 132 may retrieve data from the various databases 120 containing product information, and may produce a product-related page 125 containing the product's information or a strictly HTML page. Significantly, the product-related page 125 generated by the Seller's E-Commerce Server 132 contains "mboxes", which consist of program statements or executable codes and which will be described in greater detail below. The Seller's E-Commerce Server 132 then returns the product-related page 125 to the presentation device 34. The consumer/shopper's client renders the returned the product-related page 125, including the embedded "mboxes". Program statements or executable codes of the "mboxes" cause the consumer/shopper's client to make a request to the Mbox Server 140, sending information about the product(s) displayed. The "mboxes" may be configured such that identification of the consumer/shopper's client is sent to the Mbox Server 140 as well. The Mbox Server 140 then stores the received data into the Merchandising Database 144.

[0038] Attention now turns to embedding an mbox into a product-related page and configuring the mbox to report information. According to one embodiment of the invention, the seller can embed an mbox into a Web-page by incorporating a JavaScript function into the page's HTML codes or into a page template. The JavaScript function can also be incorporated into third-party software that dynamically generates product-related pages using data from various databases. An example JavaScript function that can be incorporated into a Web-page template is shown below in Table 1. In one embodiment, the program statements that define the procedure calls of the mbox may reside within the Web-page. In another embodiment, the program statements may also reside on the Mbox Server 140, which can be accessed by the Seller's E-Commerce Server 132 upon receiving a page request.

Table 1

```

<script language='javascript'>
mboxCreate('productPage' , 'productId=USA193' ,
5  'productName=Aviation Edition Monopoly- A Century in
Flight', 'productPageURL=' + window.location ,
'productThumbnailURL=http://us.st4.yimg.com/store4.yimg.c
om/I/boardgames_1773_6489720')</script>
<div class='mboxDefault'>
10 <!--Definition of Default Content to be displayed if no
offers are provided by the Mbox Server.-->
</div>

```

[0039] Program statements that define functions of an mbox in accordance with
15 an embodiment of the invention can be found in the Appendix. Note that, although an
embodiment of the invention is implemented using client-side JavaScript, other
embodiments of the invention may be implemented using server-side JavaScript, Java,
C++, and other programming languages.

[0040] Note that the seller can assign a name and one or more parameters to an
20 mbox. These parameters may be assigned in the form of name-value pairs. For
instance, in the example of Table 1, the seller has assigned the mbox name
"productPage" and various parameters (e.g., "productName=Aviation Edition
Monopoly- A Century in Flight", "productId=USA103") to the mbox. When the
product-related page is requested, the mbox passes the mbox name "productPage" and
25 the parameters to the Mbox Server by executing a procedure that the mbox is
programmed to perform.

[0041] According to an embodiment of the invention, the parameters assigned to
an mbox correspond to attributes of a product and links to the product's associated
digital content (images, product page links, etc.). In the example shown in Table 1,
30 the product-related page in which the mbox is embedded is a product detail page for a
Monopoly game. As such, parameters indicating attributes of the displayed product,
such as "productName=Aviation Edition Monopoly - A Century in Flight", are used.
In this way, the information reported by the mboxs will convey the product
information of the displayed product(s).

[0042] In one embodiment, some parameters provided to the Mbox Server 140 may include many different types of merchandising related information. For example, the parameters may include a description of the displayed product, which is commonly called a "copy" among merchandising professionals. Other parameters may include a client identifier that identifies the client program, and segmentation identifier that identifies the sex and age group of the shopper. Yet other parameters may include data about a product's presentation, assortment and packaging relative to other products or services. An incomplete list of exemplifying parameters that an mbox may provide to the Mbox Server 140 is shown below in Table 2.

Table 2

<u>Name</u>	<u>Value</u>
productID	USA193
productName	Aviation Monopoly ...
productPageURL	www.boardgames.com/monopoly/special_editions/...
productThumbnailURL	http://us.st4.yimg.com/store4.yimg.com/l/boardgames_1773_...
productCopy	"Presenting the Century of Flight Aviation Edition of the world's most famous board game, MONOPOLY. Buy, sell..."

[0043] The values of the parameters may be provided by a seller's server that produces the page, or may be available through the environment variables, or variables that are accessible to a web page using JavaScript or an equivalent execution environment. Examples of values that can be obtained from such means include information about a referring source, or a search query in a search results page. In one embodiment, such values are considered to be merchandising data as well.

[0044] An example product detail page showing a Monopoly game having mboxes 160a-160b is shown in FIG. 3. Messages M1 and M2, which contain information including the respective mboxes's names "ProductPageTitle" and "ProductPage" and the mbox's parameters, including "productId", "productName", and "productPageURL", are shown to be provided to the Mbox Interface 242 of the Mbox Server 140. Note that the mboxes 160a-160b are illustrated in FIG. 3 for the

purposes of depicting the invention conceptually. In practice, the Web-page does not display the mbox name or parameters to the customer.

[0045] According to one embodiment of the invention, mboxes may be configured to display promotional materials, or "offer content". In one embodiment, the Mbox Server 140 is responsible for selecting offer content to be displayed by the mboxes. In that embodiment, the seller can use the merchandising data collected by the Merchandising Database 144 to enhance conversion. For instance, the merchandising data will show which offer content is the most effective when displayed along with the Aviation Monopoly game. Armed with this information, the seller will be able to modify the offers and offer content such that the promotional materials are effectively targeted at customers according to the products they are viewing.

[0046] Note that the Merchandising Database 144 is populated without requiring data to be directly imported from product database, order database, or any other databases 120 of the Seller's E-Commerce Server 132. All the information within the Merchandising Database 144 can be obtained via the mboxes embedded throughout the product-related pages and/or check-out pages. Since the mboxes are located *in situ* on the output medium of the online catalogs and since the mboxes are rendered at the point of presentation, the information collected by the mboxes directly represents the effectiveness of various merchandising efforts. Merchandising data no longer has to be obtained from multiple disparate databases (e.g., transaction databases, purchase history databases) using costly data-mining software.

[0047] According to one embodiment of the invention, the Merchandising Database 144 can be culled periodically to remove out-dated data. In this way, the data contained in the Merchandising Database 144 will collectively represent recent browsing and/or purchasing activities of the site's visitors.

[0048] Referring now to FIG. 2, some components of the Mbox Server 140 are depicted. In particular, the Mbox Server 140 includes an Mbox Interface 242, a Logger 244, Campaign Execution Module 246, an Administration Module 222, and a Data Analysis Module 254. As shown, Mbox Interface 242, Logger 244, and Campaign Execution Module 246 are part of the runtime environment of the Mbox Server 140. The Administration Module 222 and the Data Analysis Module 254, on the other hand, are part of the administrative environment.

[0049] In one embodiment, the Mbox Interface 242 receives data collected by the mboxs and Logger 244 stores the received data into the Merchandising Database 144. In one embodiment, the Logger 244 may keep track of the number of times a product has been displayed, and stores such information within the Merchandising Database 144 together with the product information. The Campaign Execution Module 246 is responsible for the offer presentation aspect of the invention, and its operations will be discussed in greater detail below. The Administration Module 222 allows a merchandiser or a seller to define merchandising campaigns, and the Data Analysis Module 254 provides an analytic engine for the merchandising data collected in the Merchandising Database 144. Operations of those modules are also described below.

[0050] Attention now turns to operations of the Campaign Execution Module 146. In one embodiment, the Campaign Execution Module 146 monitors the information reported by the mboxs via the Mbox Interface 242. According to an embodiment of the invention, some mboxs may be used to display offers or promotional materials. When such mboxs are reporting information, the Campaign Execution Module 246 looks up appropriate offers to be displayed. The Campaign Execution Module 246 may retrieve the offers and transmit the offers back to the mboxs via the Mbox Interface 242.

[0051] According to an embodiment of the invention, some mboxs are associated with static offers. That is, the same pre-defined product information or promotional materials will be selected and provided to mboxs bearing the same mbox name during the campaign period or until it is changed. Some mboxs are associated with dynamic offers, the content of which may change according to data collected within the Merchandising Database 144. For example, an mbox may be associated with a "Top Ten Seller Offer" that specifies the display of the top-ten currently most frequently viewed products. For that mbox, the Campaign Execution Module 246 will query the Merchandising Database 144 (or access data produced by the Data Analysis Module 254) to identify the currently most frequently viewed products. The Campaign Execution Module 246 then accesses the Offer Content Server 138 and Offer Content 139 to retrieve the corresponding offer materials. Some mboxs may be associated with conditional offers. For example, an offer may specify that an mbox displays certain promotional materials if the Web-page is accessed during certain time of the day. Note that it is not necessary to assign an offer to every

mbox embedded within a Web-site. The mboxs can be configured to display a default offer when the Campaign Execution Module **246** fails to provide them with any content or promotional materials.

[0052] According to an embodiment of the invention, the Administration Module **222** may be used by the merchandiser or the seller to define a Merchandising Campaign. A Merchandising Campaign herein refers to a set of offers applicable to some or all of the mboxs embedded within a Web-site for a given period of time. Using the Administration Module **222**, the seller can define automated merchandising elements for every mbox embedded within a site.

[0053] Referring now to **FIGS. 5A-5B**, a flow diagram for a process of setting up an Mbox Server in accordance with an embodiment of the invention is shown. Part of the process has been described above in conjunction with the descriptions of **FIGS. 1** and **2**. Nevertheless, those steps are described here again in greater detail for purposes of clarity.

[0054] As shown, at step **510**, the seller embeds one or more mboxs on product detail pages and pages on which conversion events occur (e.g., check-out pages, Shopping Cart pages, order confirmation pages). As described above, an mbox may be embedded by incorporating a JavaScript function (e.g., see Table 1) into the HTML codes of a Web-page or a Web-page template used by the Web-site's servers to produce the page. Codes that define the location and size of the display area for displaying offer content can be incorporated into the HTML codes of the Web-page or Web-page template in a similar manner.

[0055] At step **520**, merchandising data is recorded within a Merchandising Database after the Web-site "goes live." When visitors to the Web-site requests pages that contain one or more mboxs (step **522**), the mboxs pass various mbox parameters to the Mbox Server **140** by executing a procedure that the mbox is programmed to perform (step **524**). The various mbox parameters and mbox names are recorded within the Merchandising Database (step **524**). During the display of pages on which conversion events occurred, the conversion event information is transmitted to the Mbox Server **140** and recorded (Step **526**). Note that conversion event information is also considered to be merchandising data. According to an embodiment of the invention, the Mbox Server **140** may modify the "offers" based on the updated information (step **528**). For example, if an offer calls for displaying the currently top selling board games, the Mbox Server **140** may send a message to the

Offer Content Server **138** to modify the Offer Content **139** whenever a transaction occurs such that the offer content reflects up-to-date purchasing trends.

[0056] At step **530**, the seller may embed more mboxs on pages where product recommendations or other offer display is desired. The mboxs that are already
5 embedded for collecting merchandising data can be selected for displaying product recommendations or other offer as well. A process of displaying product recommendations or offers via an mbox is described further below in conjunction with **FIG. 6**.

[0057] At step **540**, the seller sets up one or more Merchandising Campaigns for
10 the Web-site. This step may entail: selecting the mboxs to participate in a Merchandising Campaign (step **542**), defining the "offer" or automated merchandising element for each participating mbox (step **544**), setting the duration of the campaign (step **546**), and previewing the offers (step **548**). The Administration Module **222** (**FIG. 2**) provides a graphics user interface to allow the seller to set up the
15 Merchandising Campaigns, which is described further below with respect to **FIGS. 7** and **8**.

[0058] With reference again to **FIG. 5**, at step **550**, after the Merchandising Campaigns are defined, the selected mboxs will begin to display offers. This step may entail: an mbox requesting offer content from the Offer Content Server **138** when
20 a shopper requests a page containing the mbox (step **552**), the Offer Content Server **138** evaluating the parameters sent by the mbox and calculates appropriate products to display in the mbox (step **554**), and the shopper's Web client displaying the offers in the mbox (step **556**).

[0059] At step **560**, after the Merchandising Database has collected a
25 meaningful amount of information, various data analyses can be performed to obtain recent information about the online traffic and sales.

[0060] **FIG. 6** depicts a flow diagram for an automated merchandising process
600 in accordance with an alternate embodiment of the invention. As shown, at step **612**, the Mbox Server receives a message from an mbox. In one embodiment, the
30 message contains an mbox name and one or more parameters associated with the product(s) that the Web-page displays. This step is also illustrated for example in **FIG. 3**, which shows the Mbox Interface **242** receiving the message M2 containing the mbox name "ProductPage" and various parameters (e.g., "productId=USA193") from the mbox **160b**.

[0061] Referring again to **FIG. 6**, at step **620**, after receiving the message from the mbox, the Campaign Execution Module accesses pre-defined merchandising campaigns and offers to identify ones applicable to the mbox in question.

[0062] At step **630**, the Campaign Execution Module determines if any
5 merchandising campaign is applicable. If not, the Mbox Server retrieves a default offer from an Offer Content Database (step **645**), where offer contents are stored. In this embodiment, if there is an applicable merchandising campaign, the Campaign Execution Module retrieves an appropriate offer from the Offer Content Database (step **640**).

10 [0063] In one embodiment, the Campaign Execution Module may query the Merchandising Database for a product that meets the criteria of the applicable offer. For example, if an applicable offer calls for the display of the most frequently viewed product, the Campaign Execution Module may query the Merchandising Database to determine which products meet the requirements of the offer and may access the
15 Offer Content Server to retrieve the corresponding offer content. Additional information to be displayed by the mbox may be provided by the Mbox Server.

[0064] With reference still to **FIG. 6**, at step **295**, the Mbox Server returns the offer to the mbox for display. In one embodiment, this step is illustrated for example in **FIG. 4**, which depicts the Mbox Interface **242** sending the messages M3 and M4
20 containing offer content to the mboxes **160a-160b**. In the illustrated embodiment of **FIG. 4**, the offer content include the title "Other Customer Favorites" and pictures and links of several top-selling products.

[0065] It should be noted that the process **600** is preferably carried out after a meaningful amount of merchandising data has been collected. In one embodiment,
25 before a meaningful amount of merchandising has been collected, the Mbox Server may return a default offer.

[0066] According to an embodiment of the invention, the Administration Module **222** (**FIG. 2**) of the Mbox Server **140** provides a user-friendly GUI for the seller to set up merchandising campaigns and to define automated merchandising
30 elements for the mboxes. First, the Administration Module **222** provides pull down menus **714** that allow a merchandiser or user of the module to define the duration of the merchandising campaign. In addition, the Administration Module **222** several pull down menus **710** that allow a merchandiser or user of the module to select one or more mboxes for counting conversions and for displaying offers. In one embodiment,

these pull-down menus **710** are created from a database table maintained by the Mbox Server. The Mbox Server **140** receives requests from any mbox that is on a page that is being displayed on a shopper's client. If the Mbox Server **140** has never heard from a specific mbox with a specific mbox identifier, the Mbox Server adds the unknown mbox to the table.

[0067] Note that the mboxes may be selected by mbox names, such as "productPage" and "productPageTitle." According to an embodiment of the invention, multiple mboxes embedded within the same or different Web-sites may share the same mbox name. Thus, using the Administration Module **222**, all mboxes bearing the same mbox name may display the same offer.

[0068] Each mbox may be associated with one or more offers. As shown in **FIG. 7**, the "productPageTitle" mbox is associated with a Default Content offer and an Other Customer Favorites Offer. Pull down menu **712** allows the seller or merchandiser using the Administration Module **222** to adjust the frequency each offer is displayed. As shown, ten percent of the visitors will see Offer 1, and ninety percent of the visitors will see Offer 2.

[0069] The Administration Module **222** also provides a link **304** (**FIG. 7**) to a Browse Offer screen, which is depicted in **FIG. 8**, and which allows a user of the Administration Module **222** to modify the offers to be presented. At the Browse Offer screen, the user can choose one of the pre-defined automated product suggestions or static HTML offers to be associated with the mbox. The user may, through another user interface (not shown), define or edit the specifics of an automated product suggestions.

[0070] Attention now turns to the functions of the Data Analysis Module **254**. According to an embodiment of the invention, the Data Analysis Module **254** is responsible for performing data analysis on the data stored by the Merchandising Database **144**. In one embodiment, the Data Analysis Module **254** periodically queries the Merchandising Database **144** to obtain commonly used information (e.g., the most popular products) such that this information becomes readily available for the Campaign Execution Module **246**. In some embodiments, the Data Analysis Module **254** analyzes the merchandising data to determine item-relatedness. For example, the Data Analysis Module **254** can be programmed to query the Merchandising Database **144** and find out the products that historically purchased together. Unlike conventional approaches that utilize the user profile databases and

transaction history databases to generate a table of "similar items", the analysis performed by the Data Analysis Module 254 is less computationally intensive and does not involve direct integration of multiple databases with the analysis software.

[0071] Another function of the Data Analysis Module 254 is to generate reports that show the effectiveness of any given merchandising campaign. As an example, a merchandiser may be interested in comparing two merchandising efforts. Using an embodiment of the invention, the merchandiser may set up an mbox such that Offer A is displayed some of the times and Offer B is displayed some other times. This type of campaign is called an "A/B Comparison Campaign" herein. Throughout the campaign, appropriate parameters, such as the number of click-throughs and conversions, are recorded in the Merchandising Database 144. When the campaign is over, the Data Analysis Module 254 analyzes the collected data to determine the total number of impressions made, the number of click-throughs, and the number of conversions that occurred, etc. An example report summarizing the findings of an A/B Comparison Campaign is shown in FIG. 10. Another example report summarizing the finds of such an A/B Comparison Campaign is shown in FIG. 11. Using this report, a merchandiser can quickly and easily assess the effectiveness of one automated merchandising element vis-à-vis the other and make any fine-tuning adjustments, if necessary. It should be noted that making adjustments to the offer content no longer requires significant involvement from the Web-site's Information Technology (IT) staff. Merchandisers without significant technical or IT may use the Campaign Definition Module to effect changes in merchandising strategies.

[0072] Elements of the invention can be implemented through computer program(s) operating on one or more general purpose computer systems or instruction execution systems such as personal computers or workstations, cable TV set-top boxes, satellite TV set-top boxes, computer gaming systems, video-phone systems, mobile systems (e.g., computers, wireless telephones, personal digital assistants) or other microprocessor-based platforms. FIG. 9 illustrates details of a computer system that may be used to implement the invention. System bus 601 interconnects the major components. The system is controlled by microprocessor 602, which serves as the central processing unit (CPU) for the system. System memory 605 is typically divided into multiple types of memory or memory areas such as read-only memory (ROM), random-access memory (RAM) and others. The system memory may also contain a basic input/output system (BIOS). A plurality of general input/output (I/O)

adapters or devices 606 are present. Only three are shown for clarity. These connect to various devices including a fixed disk drive 607 a diskette drive 608, network 610, and a display 609. Computer program code instructions for implementing the functions of the invention are stored on the fixed disk drive 607. When the system is operating, the instructions are partially loaded into memory 605 and executed by microprocessor 602. Optionally, one of the I/O devices is a network adapter or modem for connection to a network, which may be the Internet. It should be noted that the system of FIG. 9 is meant as an illustrative example only. Numerous types of general-purpose computer systems are available and can be used.

10 [0073] Elements of the invention may be embodied in hardware and/or software as computer program code (including firmware, resident software, microcode, etc.). Furthermore, the invention may take the form of one or more computer program products on a computer-usable or computer-readable storage medium having computer-usable or computer-readable program code embodied in the medium for use
15 by or in connection with one or more instruction execution systems such as the one shown in FIG. 9. A computer-usable or computer-readable medium may be any medium that can contain, store, communicate, or transport the program for use by or in connection with an instruction execution system. The computer-usable or computer-readable medium can be, for example, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system. The medium may also be simply
20 a stream of information being retrieved when the computer program product is "downloaded" through a network such as the Internet. Note that the computer-usable or computer-readable medium could even be paper or another suitable medium upon which a program is printed.

25 [0074] While the invention has been described and shown in connection with the preferred embodiments, it is to be understood that modifications may be made without departing from the spirit thereof. The embodiments described are by way of example and should not be construed as limiting of the claims except where referenced to the specification is required for such construction. For instance, it
30 should also be understood that throughout this disclosure, where a software process or method is shown or described, the steps of the method may be performed in any order or simultaneously, unless it is clear from the context that one step depends on another being performed first. It should be understood by those skilled in the art upon reading the present disclosure that software processes, which have been described as client-

side processes (e.g., those running on the presentation devices), can be performed as server-side processes (e.g., those running on a server), and vice versa, when appropriate.

APPENDIX

```
//
5 // $Id: mbox.jsp,v 1.7 2004/02/20 02:24:24 mcyster Exp $
//
// Copyright 2003 Fort Point Partners, Inc. All Rights Reserved.
//
// This software is the proprietary information of Fort Point
10 Partners,
Inc.
// Use is subject to license terms.
//

15 if (typeof mboxIncluded == 'undefined') {
    var mboxIncluded = true;

    var mboxIsOnMac = window.navigator.appVersion.indexOf("Mac") != -1
    var mboxIsOnIEMac = document.all && mboxIsOnMac
20 var mboxIsOnIE4 = document.all &&
    (window.navigator.appVersion.indexOf("MSIE 4.") != -1)
    var mboxIsOnNS4 = (navigator.appName == 'Netscape') &&
    (parseInt(navigator.appVersion) == 4);

25 var mboxSupportedPlatform = true;
    if (mboxIsOnNS4 || mboxIsOnIE4) {
        mboxSupportedPlatform = false;
    }

30 var mboxIsEnabled = mboxSupportedPlatform;
    var mboxIsSetup = false;
    var mboxTimeout = 15000;

35 var mboxClientCode = 'fortpointpartners';

40 var mboxServerURL =
'http://mbox.fortpoint.com/boxserver/content.js';

45 var mboxCookieDomain = ""; // set if needed e.g. "
domain=fortpoint.com"
    var mboxClickedCookie = "mboxClick";
    var mboxSessionIdCookie = "mboxSession";
    var mboxPCIdCookie = "mboxPC";
50 var mboxDisableCookie = "mboxDisable";
    var mboxDisableCookieDuration = 60 * 60;
    var mboxCheckCookie = "mboxCheck";
    var mboxDebugModeCookie = "mboxDebug";

55 var mboxNameArg="mbox";
    var mboxClientCodeArg="mboxClient";
    var mboxPCIdArg="mboxPC";
    var mboxSessionIdArg="mboxSession";
    var mboxPageIdArg="mboxPage";
60 var mboxHostArg="mboxHost";
    var mboxURLArg="mboxURL";
```

```

var mboxReferrerArg="mboxReferrer";

var mboxMarkerDivPrefix="mboxMarker-";
var mboxImportedDivPrefix="mboxImported-";
5
var mboxPageId = mboxGenerateId();

var mboxes = new Array(); // array of object mbox
var mboxGlobalArguments = "";
10
if (mboxSupportedPlatform) {
    var mboxOnloadFunctions = new Array();

    mboxAddOnloadFunction(mboxCheckAll);
15
    if (mboxDebugMode()) {
        mboxAddOnloadFunction(mboxDebugInfo);
    }

    mboxSignalClicked();
20
    if (mboxEnabled()) {
        mboxDefaultContentDisplayNone();
    }
    else {
25
    }
    } else {
        if (mboxDebugMode()) {
            alert("mbox functionality is not supported on this browser");
            mboxDeleteCookie(mboxDebugModeCookie);
30
        }
    }
}

35
/*
 * Puts a mbox called 'mboxId' on the page.
 *
 * Parameters:
 *   mboxId - Id of this mbox
40
 *   ... - 0 or more arguments of the form
 *         variableName=value or
variableName1=value1&variableName2=value2
 *         (the values need to be escaped for inclusion in the url)
 */
45
function mboxCreate(mboxId /*, ... */) {
    if (!mboxSupportedPlatform) {
        return;
    }

50
    var id = mboxId;
    arguments[0] = mboxNameArg + '=' + id;
    var box = new mbox(id, mboxBuildURL(arguments));
    box.put();
}

55
/*
 * Stores its parameters in the global parameter array which
 * will be added to each mbox url.
 */
60
function mboxGlobalParameters(/* ... */) {
    for (var i = 0; i < arguments.length; i++) {

```

```

        mboxGlobalArguments = mboxGlobalArguments + "&" + arguments[i];
    }
}

5  /*
   * mbox constructor
   *   id - id of mbox
   *   url - full url to fetch mbox from server
   */
10 function mbox(id, url) {
    this.id = id;

    this.url = url;
    this.timeout = null;

15    this.put = mbox_put;
    this.hide = mbox_hide;
    this.startTimeout = mbox_startTimeout;
    this.cancelTimeout = mbox_cancelTimeout;
20    this.showDefaultContentIfNotOk = mbox_showDefaultContentIfNotOk;
    this.hasDefaultContent = mbox_hasDefaultContent;
    this.hideDefaultContent = mbox_hideDefaultContent;
    this.showDefaultContent = mbox_showDefaultContent;
}

25  /*
   * Creates an mbox given its id and the complete mbox Url
   */
function mbox_put(timeout) {
30    mboxes[mboxes.length] = this;

    if (mboxEnabled()) {
        document.write(
            '<div id="' + mboxMarkerName(this.id) + '">'
35            + '<script language="JavaScript">mboxStartTimeoutById("'
              + this.id + '", ' + mboxTimeout + ');<' + '\\script>'
              + '<script src="' + this.url + '" language="JavaScript"><' +
            '\\script>'
              + '</div>');
40    }
    else {
        document.write('<div id="' + mboxMarkerName(this.id) +
            '"></div>');
    }
45 }

function mbox_showDefaultContentIfNotOk() {
    this.cancelTimeout();
    if (document.getElementById(mboxImportName(this.id)) != null) {
50        this.hideDefaultContent();
    } else {
        this.hide();
        this.showDefaultContent()
    }
}

55 }

function mbox_hide(mboxId) {
    var span = document.getElementById(mboxMarkerName(this.id));
    if (span != null) {
60        span.style.visibility = "hidden";
        span.style.display = "none";
    }
}

```



```

    }
}

function mbox_hasDefaultContent() {
5   return document.getElementById(mboxMarkerName(this.id)) != null;
}

function mbox_hideDefaultContent() {
10  mboxHideDefaultContent(this.id);
}

function mbox_showDefaultContent() {
    mboxShowDefaultContent(this.id);
}

15 function mbox_startTimeout(duration) {
    this.timeout =
        setTimeout('mboxCheckTimeoutById("'" + this.id + "'", duration);
}

20 function mbox_cancelTimeout() {
    if (this.timeout != null) {
        clearTimeout(this.timeout);
    }
}

25 /*
   * Appends the passed args to the url
   */
30 function mboxBuildURL(args) {
    var result = mboxServerURL;

    if (document.location.protocol == "https:") {
        result = result.replace("http:", "https:");
35    }
    result = result + "?" + mboxClientCodeArg + "=" + mboxClientCode;
    result = result + "&" + mboxHostArg + "=" +
document.location.hostname;

40    for (var i = 0; i < args.length; i++) {
        result = result + "&" + args[i];
    }

    result = result + mboxGlobalArguments;

45    if (result.indexOf(mboxSessionIdArg) == -1) {
        result = result + "&" + mboxSessionIdArg + "=" + mboxSessionId();
    }

50    return result + "&" + mboxPCIdArg + "=" + mboxPCId()
        + "&" + mboxPageIdArg + "=" + mboxPageId
        + "&" + mboxURLArg + "=" + escape(document.location)
        + "&" + mboxReferrerArg + "=" + escape(document.referrer);
}

55 /*
   * return true if mboxses are enabled
   */
function mboxEnabled() {
60    if (mboxIsSetup) {
        return mboxIsEnabled;
    }
}

```

```

    }
    mboxIsSetup = true;

    if (!mboxIsEnabled) {
5      return false;
    }

    // No cookie => no click through and timeout continuously redirects.
    if (!mboxCookieIsEnabled()) {
10      mboxIsEnabled = false;
      return false;
    }

    if (mboxGetCookie(mboxDisableCookie) == "true") {
15      mboxIsEnabled = false;
      return false;
    }

    return mboxIsEnabled;
20 }

function mboxStartTimeoutById(id, duration) {
    for (var i = 0; i < mboxs.length; i++) {
        if (mboxs[i].id == id) {
25          mboxs[i].startTimeout(duration);
        }
    }
}

30 function mboxCancelTimeoutById(id) {
    for (var i = 0; i < mboxs.length; i++) {
        if (mboxs[i].id == id) {
            mboxs[i].cancelTimeout();
        }
35    }
}

function mboxCheckTimeoutById(id) {
    if (document.getElementById(mboxImportName(id)) == null) {
40      // reload page with mboxs disabled
      mboxSetCookie(mboxDisableCookie, "true",
mboxDisableCookieDuration);
      // there is a risk the user will be prompted to resubmit a form
      window.location.reload(false);
45    }
}

/*
 * Checks all mboxs have been successfully imported
50 * and shows default content for any that failed
 */
function mboxCheckAll() {
    for (var i = 0; i < mboxs.length; i++) {
55      mboxs[i].showDefaultContentIfNotOk()
    }
}

/*
 * return the default content div for 'mboxName'
60 */
function mboxGetDefaultDiv(mboxName) {

```

```

    var div = mboxGetNextDefaultDiv(
        document.getElementById(mboxMarkerName(mboxName)));
    return div;
}
5
/*
 * return the next DIV tag to 'currentMbox' of class 'mboxDefault'
 */
function mboxGetNextDefaultDiv(currentMbox) {
10     if (currentMbox == null) {
        return null;
    }

    var node = currentMbox.nextSibling;
15     if (node == null) {
        return null;
    }
    if ((node.nodeType == 1) // ELEMENT_NODE type
        && (node.nodeName == "DIV")
        && (node.className == "mboxDefault")) {
20         return node;
    } else {
        return mboxGetNextDefaultDiv(node)
    }
25 }

/*
 * The name used in the div written to mark the location of the mbox
 */
30 function mboxMarkerName(mboxId) {
    return mboxMarkerDivPrefix + mboxId;
}

/*
35 * The name used in the id of the div written remotely by the mbox
    server
    */
function mboxImportName(mboxId) {
40     return mboxImportedDivPrefix + mboxId;
}

/*
 * Toggles default content visibility for the specified mbox.
 */
45 function mboxToggleDefaultContent(mboxName, visibility, display) {
    var div = mboxGetDefaultDiv(mboxName);
    if (div != null) {
        div.style.visibility = visibility;
        div.style.display = display;
50     }
}

/*
 * Shows default content visibility for the specified mbox.
 */
55 function mboxShowDefaultContent(mboxName) {
    mboxToggleDefaultContent(mboxName, "visible", "block");
}

60 /*
 * Hides default content visibility for the specified mbox.

```

```

    */
    function mboxHideDefaultContent(mboxName) {
        mboxToggleDefaultContent(mboxName, "hidden", "none");
    }
5
    /*
    * Called from the imported div tag to signal that
    * a mbox was clicked on
    */
10
    function mboxClicked(url)
    {
        mboxSetSessionCookie(mboxClickedCookie, url);
        return false;
    }
15
    /*
    * Create clicked mbox if there exists a clicked mbox cookie
    */
    function mboxSignalClicked()
20
    {
        var clickedURL= mboxGetCookie(mboxClickedCookie);

        if (clickedURL != null && clickedURL != "") {
            mboxSetSessionCookie(mboxClickedCookie, "");
25
            mboxCreateClickedBox(clickedURL);
        }

        return false;
    }
30
    /*
    * Create a clicked mbox given the associated mbox url
    */
    function mboxCreateClickedBox(url) {
35
        start = url.indexOf(mboxNameArg + "=");
        if (start == -1) {
            document.write("<!-- mboxCreateClickedBox(" + url + ") - error
-->");
            return
40
        }
        end = url.indexOf("&", start)
        if (end == -1) {
            end = url.length;
        }
45
        clickedId = url.substring(start + mboxNameArg.length + 1, end) +
        "--clicked";

        url = url.replace(url.substring(start, end), mboxNameArg + "=" +
        clickedId);
50
        var clickedBox = new mbox(clickedId, url);

        document.write('<div style="visibility:hidden;display:none">');
        clickedBox.put();
        document.write('</div>');
55
    }

    /*
    * Causes all mbox default content to be not displayed:
    */
60
    function mboxDefaultContentDisplayNone() {
        document.write("<style>.mboxDefault{ display:none; }</style>");
    }

```

```

    }

    /*
    * Builds a mbox argument, escaping the value
5   * Arguments should really be escaped by the page templating language
    */
    function mboxArg(variableName, value) {
        return variableName + "=" + escape(value);
    }
10
    /*
    ** Generates a random sessionId for the user
    */
    function mboxSessionId()
15  {
        var sessionId = mboxGetCookie(mboxSessionIdCookie);

        if (sessionId == null || sessionId.length == 0) {
            sessionId = mboxGenerateId();
20         mboxSetSessionCookie(mboxSessionIdCookie, sessionId);
        }

        return sessionId;
    }
25
    function mboxPCId() {
        var pcId = mboxGetCookie(mboxPCIdCookie);

        if (pcId == null || pcId.length == 0) {
30         pcId = mboxGenerateId();
            mboxSetCookie(mboxPCIdCookie, pcId, 2 * 365 * 24 * 60 * 60);
        }

        return pcId;
35  }

    function mboxGenerateId()
    {
        return (new Date().getTime()) + "-" + Math.floor(Math.random() *
40  999999);
    }

    /*
    * return true if the user has requested debug mode in the url
    */
45  function mboxDebugMode() {
        var result = false;

        if (document.location.href.indexOf(mboxDebugModeCookie + '=') != -
50  1)
        {
            result = true;
            mboxSetSessionCookie(mboxDebugModeCookie, "1")
        }
55
        return result || mboxGetCookie(mboxDebugModeCookie) == "1";
    }

    function mboxDebugInfo() {
60     var window = new mboxDebugWindow();
        window.put("<p>");

```

```

        window.put(" <b>Page</b>: " + document.location);
        window.put(" <ul>");
        window.put(" <li>Enabled: " + mboxEnabled() + "</li>");
        window.put(" <li>Cookies enabled: " + mboxCookieIsEnabled() +
5  "</li>");
        window.put(" <li>Global arguments: '" + mboxGlobalArguments
            + "'</li>");
        window.put(" <li>Page Parameters: '" + document.location
            + "'</li>");
10  window.put(" <li>Cookies: '" + document.cookie + "'</li>");

        for (var i = 0; i < mboxes.length; i++) {
            window.put(" <li>mbox: '" + mboxes[i].id + "'");
15  window.put(" <ul>");

            var mbox = document.getElementById(mboxMarkerName(mboxes[i].id));
            if (mbox == null) {
                window.put(" <li>error: can't find mbox in dom</li>");
20  }
            else {
                window.put(" <li>has default content: "
                    + mboxes[i].hasDefaultContent(mbox) + "</li>");
            }
25  window.put(" <li><a href=\"\" + mboxes[i].url + "\">\"
            + mboxes[i].url + "</a></li>");
            window.put(" </ul>");
            window.put(" </li>");
        }
30  window.put(" </ul>");
        window.putCloseButton()
        window.put("</p>");
    }

35  function mboxDebugWindow() {
        var name = document.location.hostname;
        var notAlphaNumeric = /\W/g
        var safeName = name.replace(notAlphaNumeric, '_');
40

        this.put = mboxDebugWindow_put;
        this.putCloseButton = mboxDebugWindow_putCloseButton;
        this.window = window.open("", "mboxDebugWindow" + safeName,
            "width=600,height=300,resizable,scrollbars=yes,toolbar=yes");
45

        if (this.window == null) {
            alert("Unable to open M2's mboxDebugWindow Window.\n"
                + "Are you blocking popups?\n");
            return;
50  }

        if (this.window.document.getElementById("mboxDebugWindowSetup") ==
            null) {
            this.put("<html><head>\n")
15  this.put(" <title>Mbox Debug Window - " + name + "</title>\n");
            this.put(" <script>");
            this.put(" function mboxWindowClose() {}");
            this.put(" document.cookie = \"\" + mboxDebugModeCookie
                + "=0; expires=Thu, 01-Jan-70 00:00:01 GMT; path=/\";");
60  this.put(" window.close();");
            this.put(" }");

```

```

        this.put(" </" + "script>");
        this.put(" </head>\n");
        this.put(" <body>\n");
        this.put(" <div id=\"mboxDebugWindowSetup\"></div>\n");
5    }
    }

    function mboxDebugWindow_put(string) {
        if (this.window == null) {
10        return;
        }
        this.window.document.writeln(string);
        this.window.scrollTo(0, 1000);
    }

15    function mboxDebugWindow_putCloseButton() {
        this.put("<a href=\"javascript:mboxWindowClose()\">"
            + "click here to close debug window</a>");
    }

20    /*
    ** Allows multiple onload functions to be supported
    ** by the one Window Onload hook
    */
25    function mboxAddOnloadFunction(f) {
        if (mboxIsOnIEMac && mboxIsOnIE4) {
            // Any current Onload functions is lost
            mboxOnloadFunctions[mboxOnloadFunctions.length] = f;
            window.onload = mboxSafeOnload;
30        } else if (window.onload) {
            if (window.onload != mboxSafeOnload) {
                mboxOnloadFunctions[mboxOnloadFunctions.length] =
                    window.onload;
                window.onload = mboxSafeOnload;
35            }
            mboxOnloadFunctions[mboxOnloadFunctions.length] = f;
        } else {
            window.onload = f;
        }
40    }

    function mboxSafeOnload() {
        for (var i = 0; i < mboxOnloadFunctions.length; i++) {
45        }
    }

    /*
    * Set cookie 'name' to 'value' for 'duration' seconds
50    */
    function mboxSetCookie(name, value, duration) {
        var expiration = new Date();
        expiration.setTime(expiration.getTime() + duration * 1000);
        document.cookie = name + "=" + escape(value) + "; expires="
55        + expiration.toGMTString() + "; path=/" + mboxCookieDomain;
    }

    /*
60    * Set a session cookie 'name' to 'value'
    */

```

```
function mboxSetSessionCookie(name, value)
{
    document.cookie = name + "=" + escape(value) + "; path=/" +
mboxCookieDomain;
5 }

/*
 * Get the value of cookie 'name'
 */
10 function mboxGetCookie(name)
{
    var value = null;

    var cookieStart = document.cookie.indexOf(name + "=");
15 if (cookieStart == -1) {
        return null;
    }
    var valueStart = cookieStart + name.length + 1;

20 var valueEnd = document.cookie.indexOf(";" , valueStart);
    if (valueEnd == -1) {
        valueEnd = document.cookie.indexOf(",", valueStart);
        if (valueEnd == -1) {
25         valueEnd = document.cookie.length;
        }
    }

    return unescape(document.cookie.substring(valueStart, valueEnd));
}
30

/*
 * Delete cookie 'name', with or without a value
 */
function mboxDeleteCookie(name) {
35     var expiration = new Date();
    var time = expiration.getTime() - 365 * 24 * 60 * 60 * 1000;
    if (time < 0) {
        time = 0;
    }
40     expiration.setTime(time);
    document.cookie = name + "=" + expires=" + expiration.toGMTString()
        + "; path="/;
    document.cookie = name + "; expires=" + expiration.toGMTString()
        + "; path="/;
45 }

function mboxCookieIsEnabled() {
    mboxSetSessionCookie(mboxCheckCookie, "true");
    return mboxGetCookie(mboxCheckCookie) == "true";
50 }
```


WHAT IS CLAIMED IS:

1. A method of populating a merchandising product database, comprising:
obtaining merchandising data related to a product from a point of presentation
5 of the product; and
storing at least part of the obtained merchandising data in the merchandising
product database.
2. The method of claim 1, wherein the obtaining step comprises obtaining the
10 merchandising data directly from a presentation device on which the product is
presented.
3. The method of claim 1, wherein the obtaining step comprises obtaining data
about the product directly from the point of presentation of the product.
15
4. The method of claim 1, wherein the merchandising product database does not
have information related to the product stored therein prior to the storing step.
5. The method of claim 1, wherein the obtaining step comprises obtaining the
20 merchandising data generally contemporaneously with presentation of the product.
6. The method of claim 1, wherein the point of presentation comprises an output
medium of an interactive catalog.
- 25 7. The method of claim 1, wherein the point of presentation comprises a Web-
page.
8. A merchandising database system, comprising:
an interface configured to coupled to a network and to receive merchandising
30 data related to a product from a point of presentation of the product; and
a database configured to store at least part of the received merchandising data.

9. The system of claim 8, wherein the interface is configured to receive the merchandising data directly from a presentation device on which the product is presented.
- 5 10. The system of claim 8, wherein the interface is configured to receive data about the product directly from the point of presentation of the product.
11. The system of claim 8, wherein the interface is configured to receive the merchandising data generally contemporaneously with presentation of the product.
- 10 12. The system of claim 8, wherein the point of presentation comprises an output medium of an interactive catalog.
13. The system of claim 8, wherein the point of presentation comprises a Web-
15 page.
14. A merchandising database system, comprising:
means for receiving merchandising data related to a product from a point of presentation of the product; and
20 means for storing the received merchandising data in a merchandising product database.
15. The system of claim 14, wherein the means for receiving comprises means for obtaining the merchandising data directly from a presentation device on which the
25 product is presented.
16. The system of claim 14, wherein the means for receiving comprises means for obtaining data about the product directly from the point of presentation of the product.
- 30 17. The system of claim 14, wherein the means for receiving comprises means for obtaining the merchandising data generally contemporaneously with presentation of the product.

18. The system of claim 14, wherein the point of presentation comprises an output medium of an interactive catalog.

19. The system of claim 14, wherein the point of presentation comprises a Web-
5 page.

20. A computer program product for use in conjunction with a computer device, the computer program product comprising a computer usable medium and a computer program mechanism embodied therein that enables the computer device to perform a
10 method of populating a merchandising product database, the computer program mechanism comprising:

first computer program codes to cause the computer device to obtain merchandising data related to a product from a point of presentation of the product;
and

15 second computer program codes to cause the computer device to store at least part of the obtained merchandising data in the merchandising product database.

21. The computer program product of claim 20, wherein the first computer program codes comprise computer program codes to cause the computer device to
20 obtain the merchandising data directly from a presentation device on which the product is presented.

22. The computer program product of claim 20, wherein the first computer program codes comprise computer program codes to cause the computer device to
25 obtain data about the product directly from the point of presentation of the product.

23. The computer program product of claim 20, wherein the first computer program codes comprise computer program codes to cause the computer device to
30 obtain the merchandising data generally contemporaneously with presentation of the product.

24. The computer program product of claim 20, wherein the point of presentation comprises an output medium of an interactive catalog.

25. The computer program product of claim 20, wherein the point of presentation comprises a Web-page.

26. A method of populating a merchandising product database, comprising:
5 rendering at least a portion of an interactive catalog, wherein content of the rendered portion includes description of one or more products and wherein the content is obtained from one or more source product databases;
communicating data related to the products to the merchandising product
database such that data is communicated from the source product databases to the
10 merchandising product database without requiring direct data importation.

27. The method of claim 26, wherein the rendered portion of the interactive catalog acts as a conduit through which information from the source product databases is communicated to the merchandising product database.

15

28. The method of claim 26, wherein the data related to the products comprises parameters embedded within the rendered portion of the interactive catalog.

29. The method of claim 26, wherein the data related to the products comprises a
20 product identification.

30. The method of claim 26, wherein the data related to the products comprises a product description.

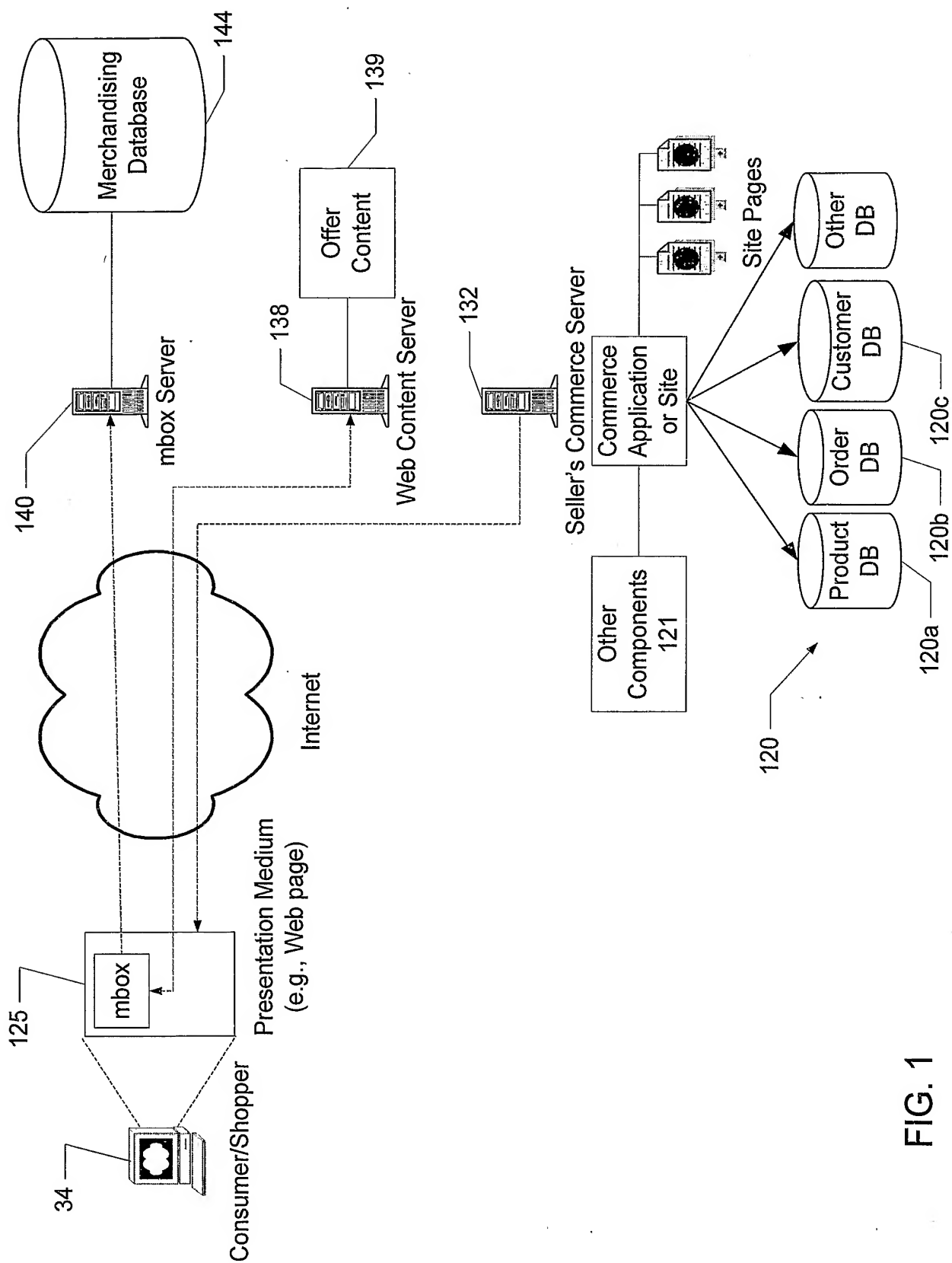


FIG. 1

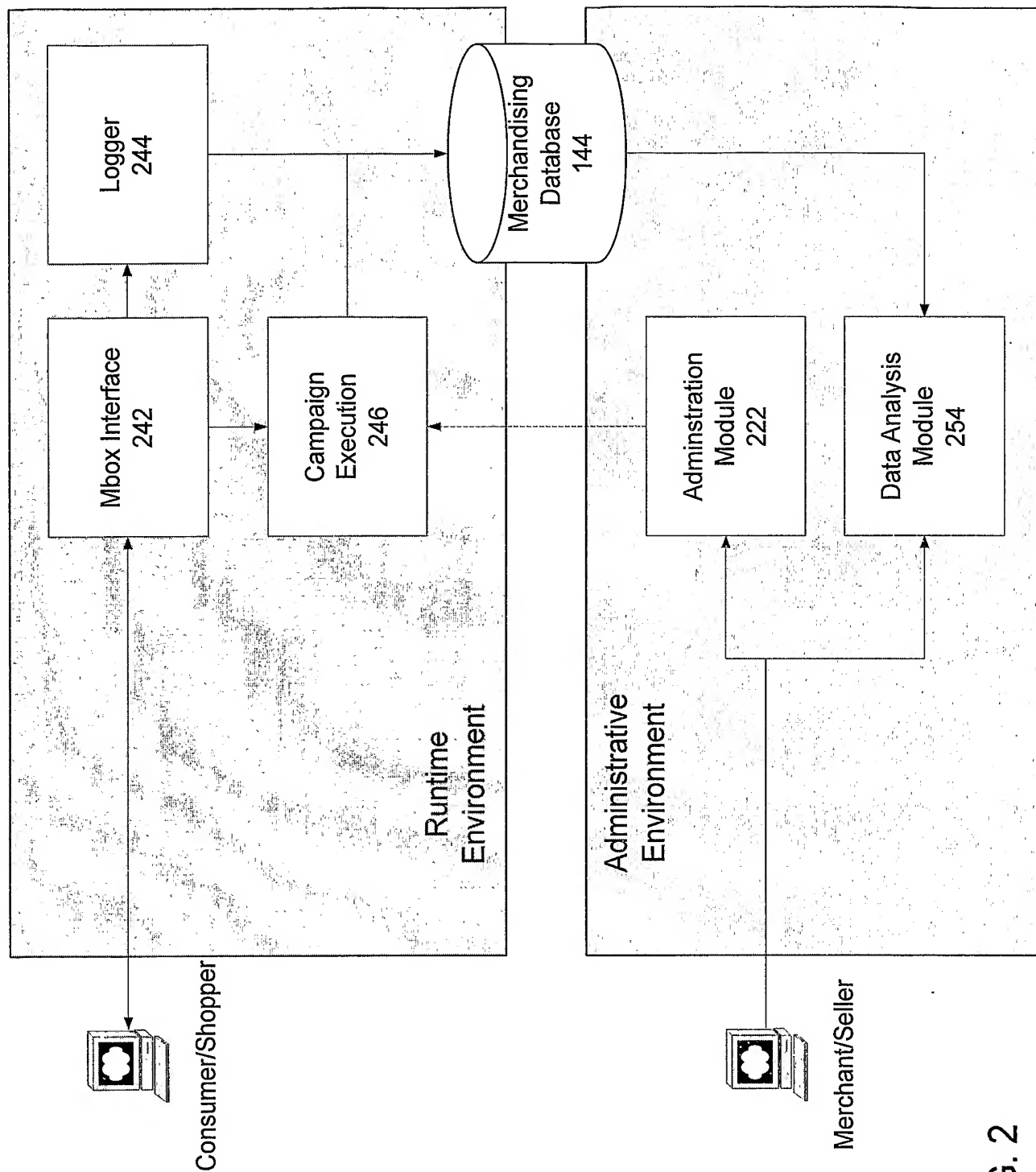


FIG. 2

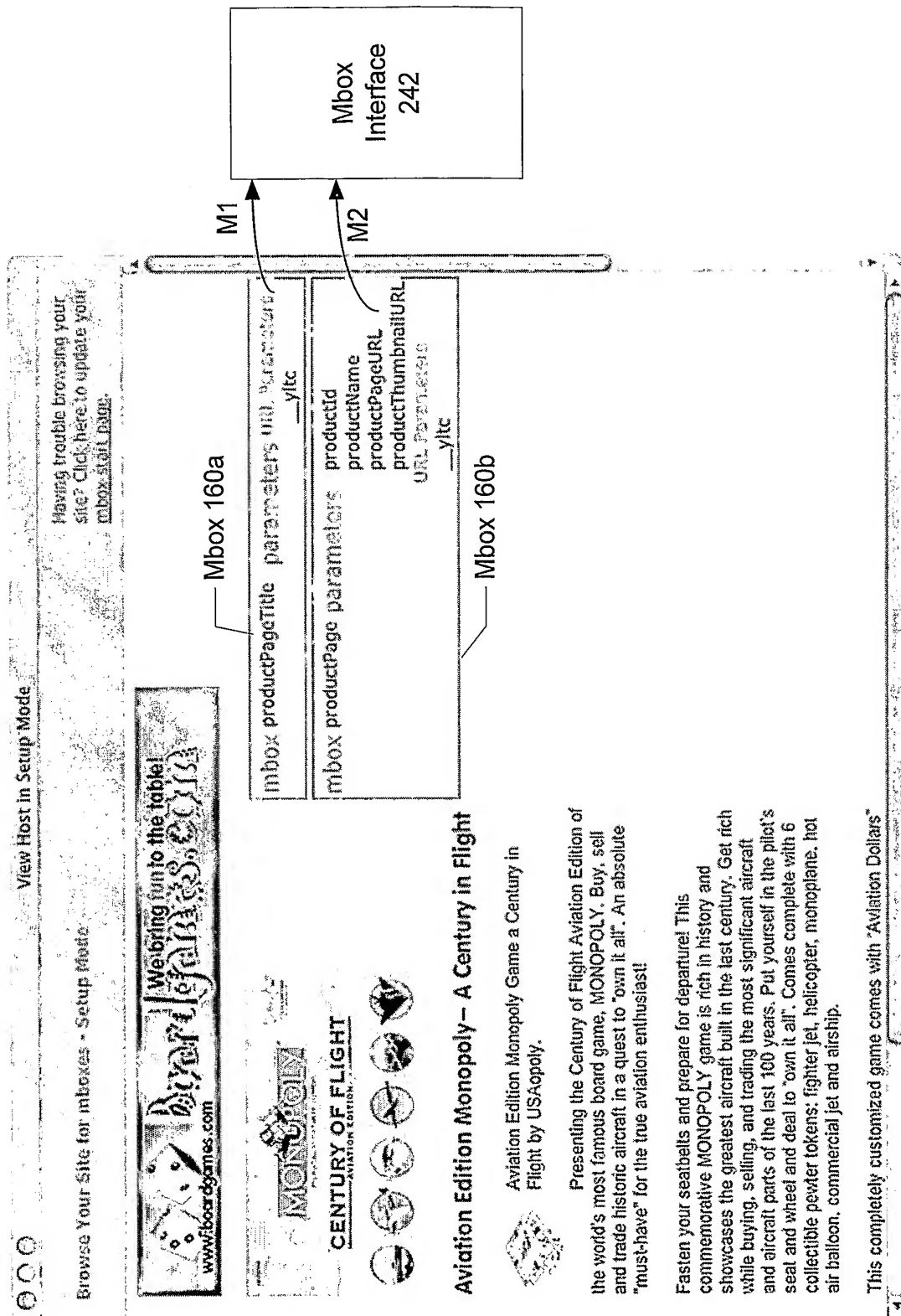


FIG. 3

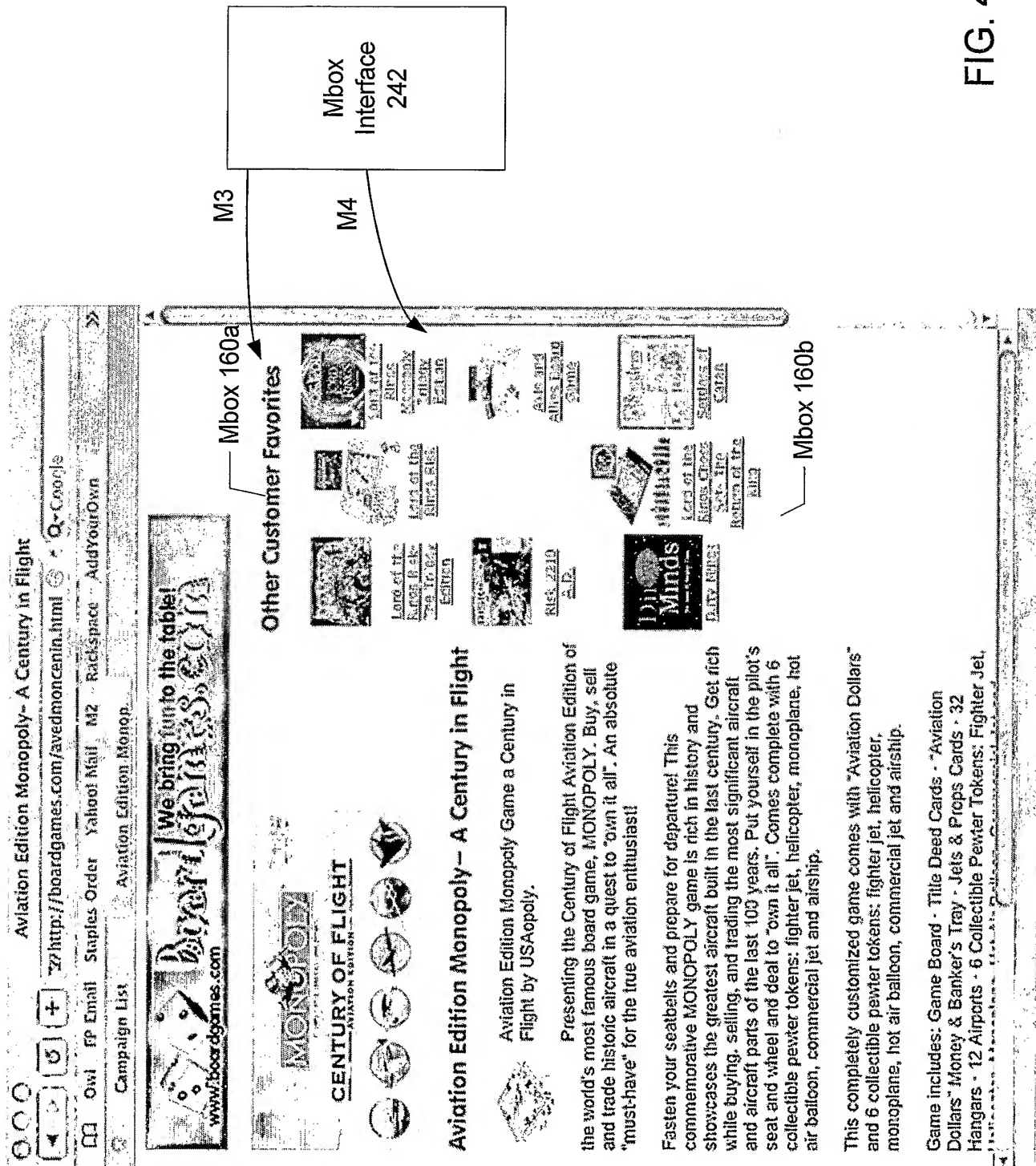
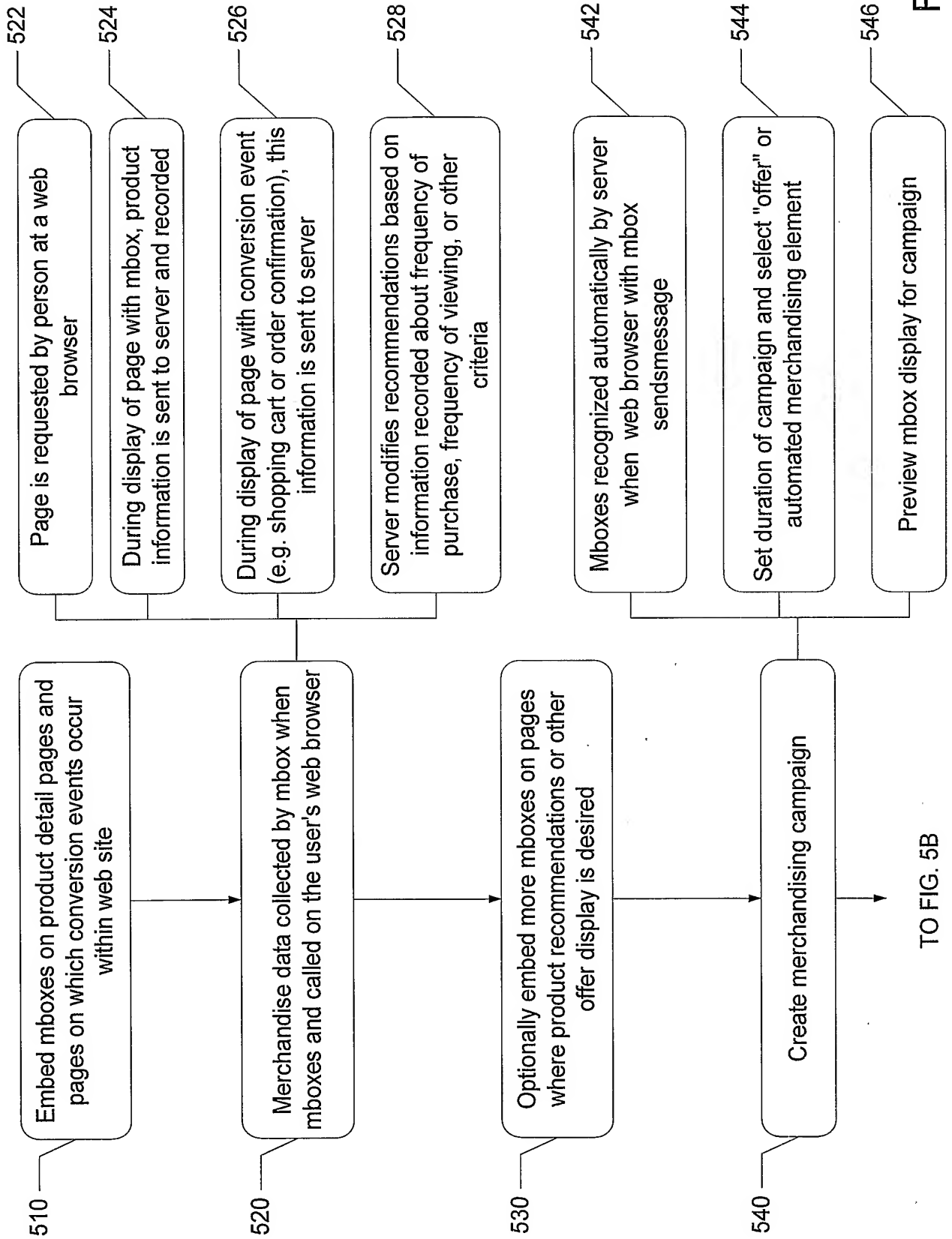


FIG. 4



TO FIG. 5B

FIG. 5A

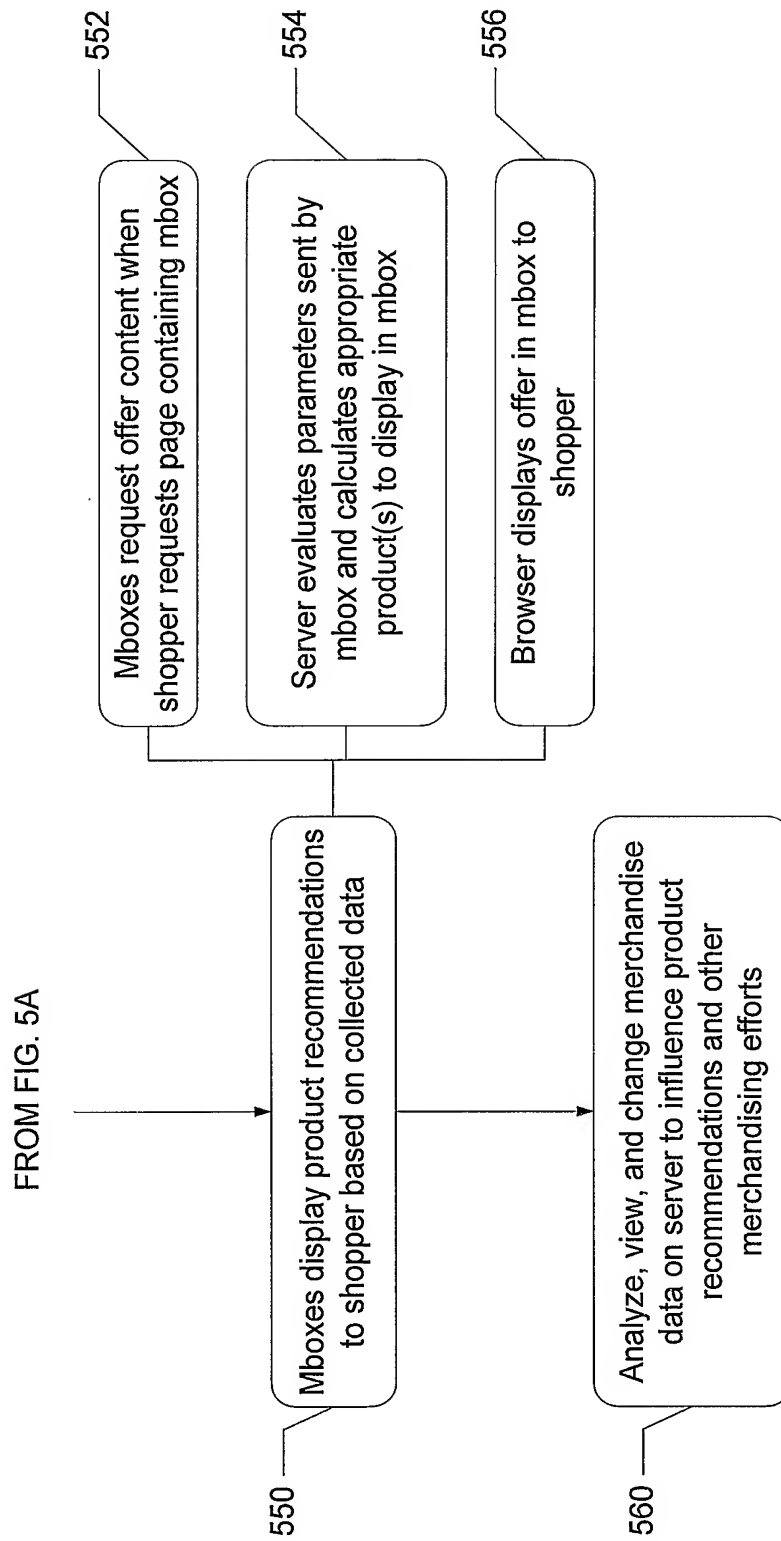


FIG. 5B

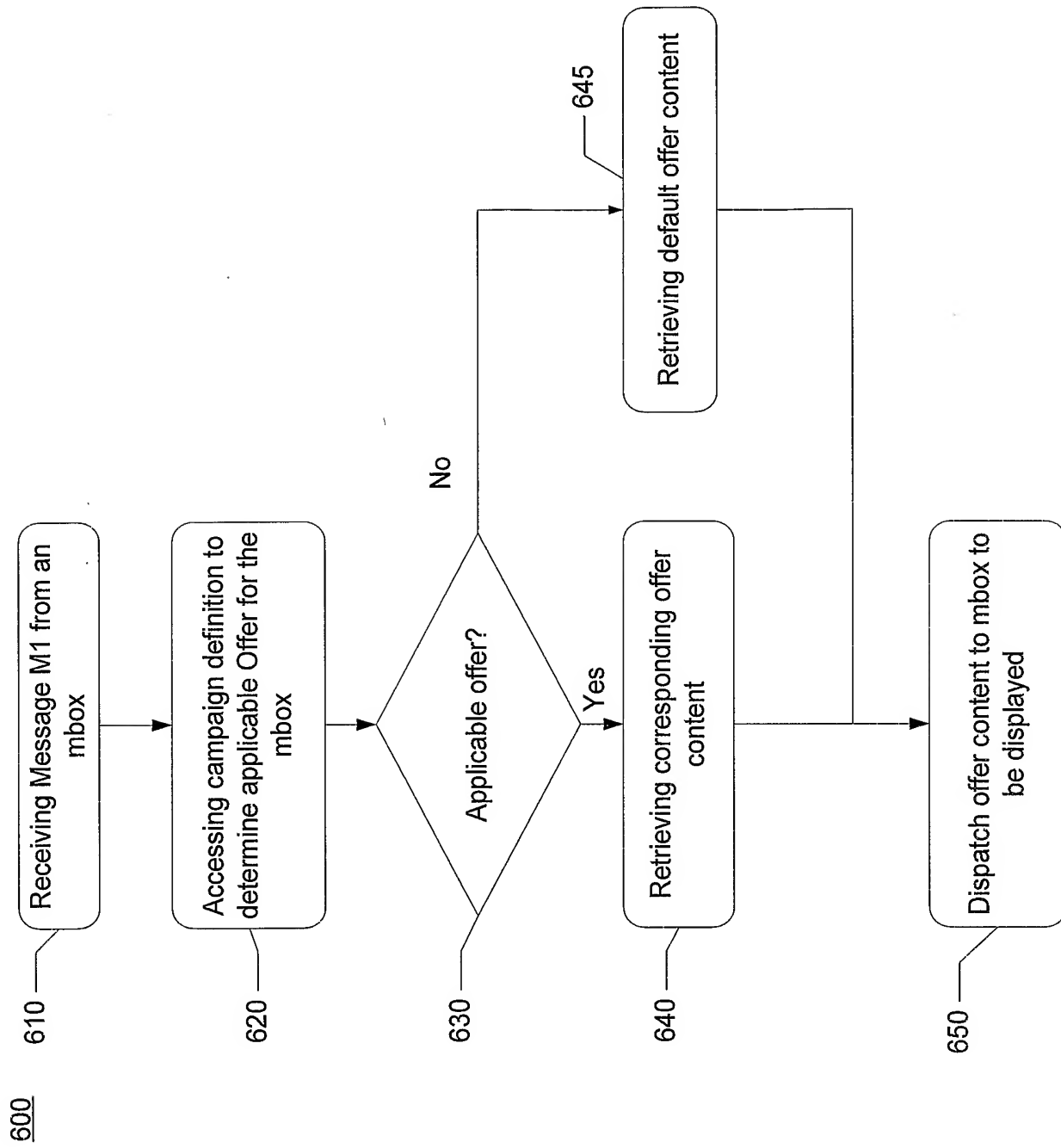


FIG. 6

Test Offers

http://m2.fortpoint.com/admin/home.do?action=edit&id=125

Test Offers

When the campaign should run

Start: 1 6 2004

End: 1 6 2005

What to name the campaign

90 - 10 Split Top Viewed on Product Page

10 of visitors will see Offer 1

Where to show your offer

Product Page Title

What offer to show

Default Content (Offer 1)

Other Customer Favorites (Offer 2)

Default Content (Offer 1)

1 Week Top Nine Products (Offer 2)

Which mbox records conversion

Which values are important

orderConfirmPage

Specify values to match for

How important the campaign is

Priority Low

save save and preview cancel

new mbox and offer

FIG. 7

Test Offers

☐ Owl IP Email
 ☐ Staples Order
 ☐ Yahoo! Mail
 ☐ M2 Rackspace
 ☐ AddYourOwn
 ☐ Google Adw.
 ☐ Word Tools
 ☐ Enter Bug

ॐ
ॐ
ॐ

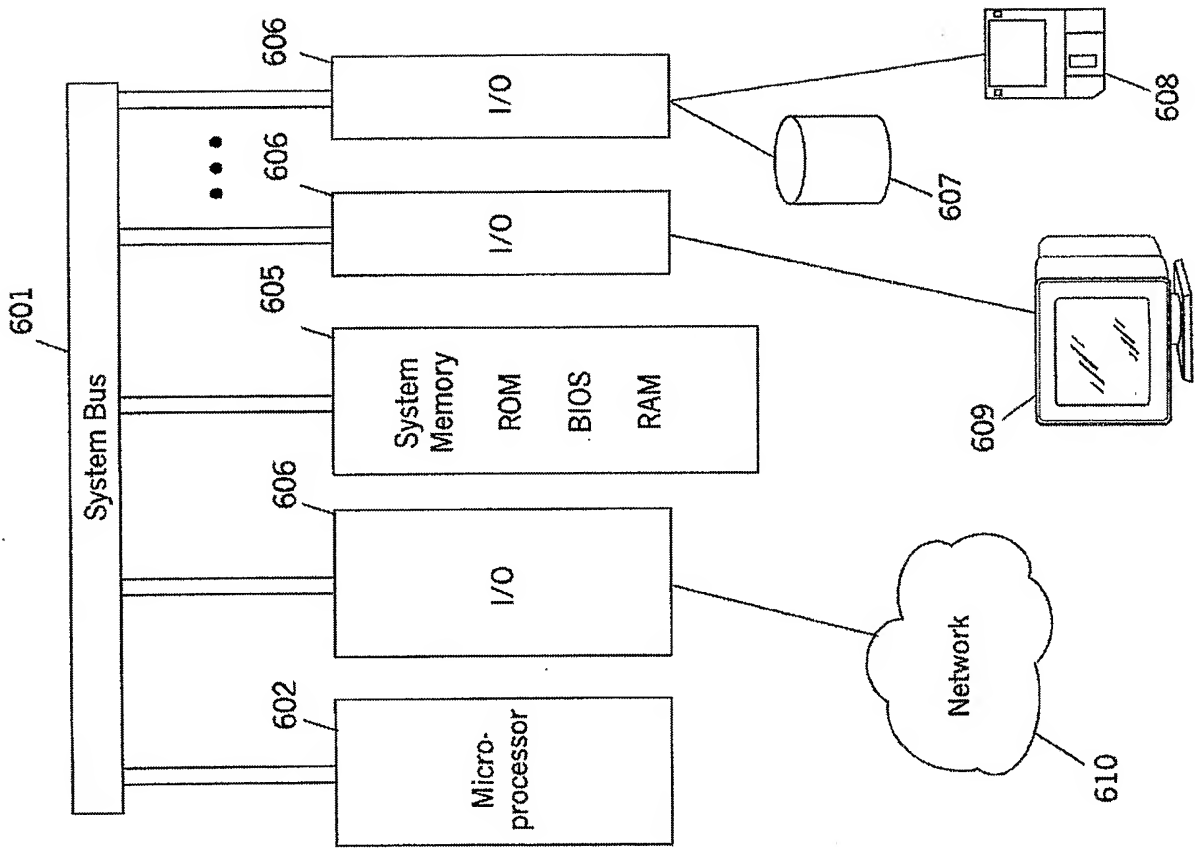


FIG. 9

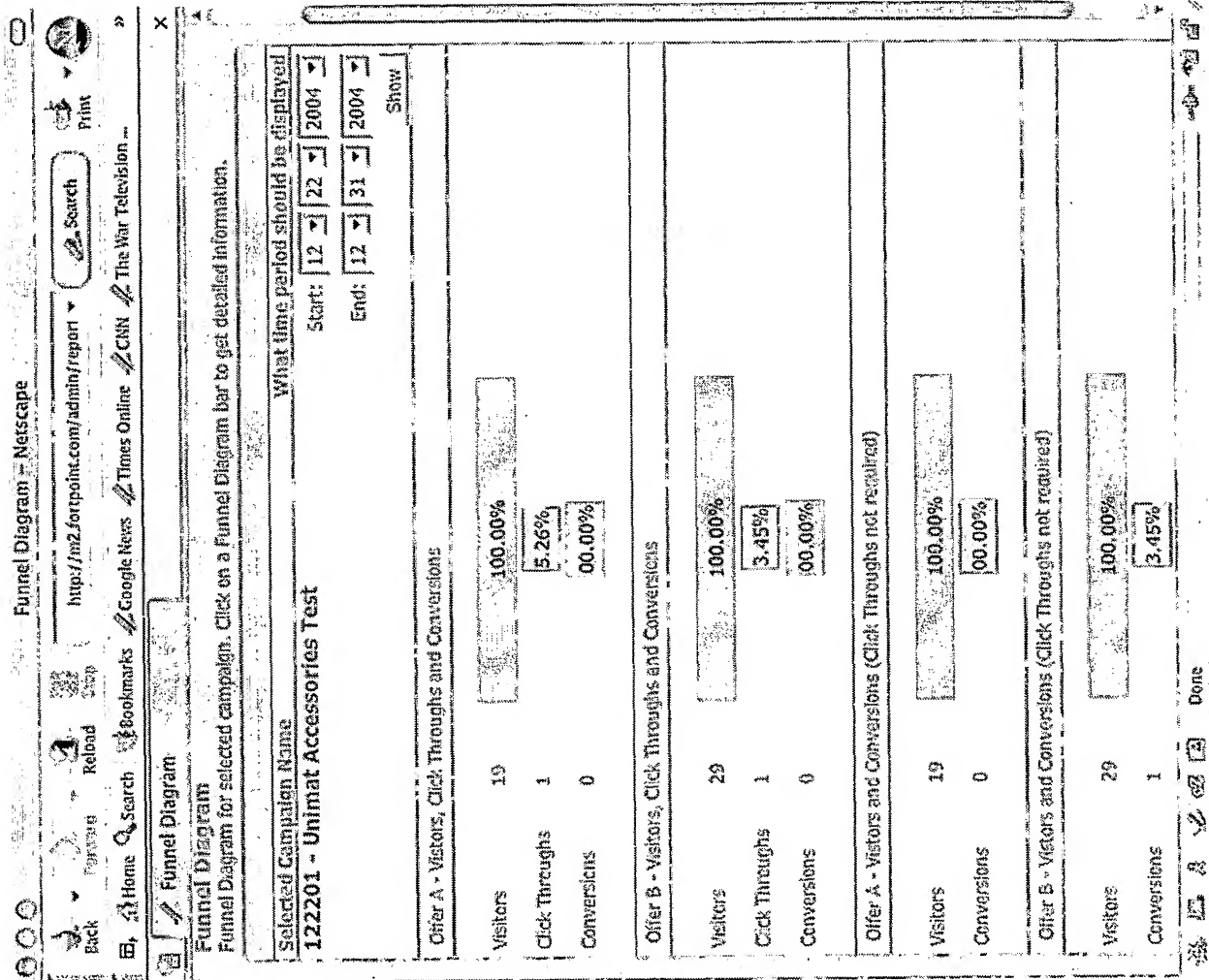


FIG. 10

